MITS®
Micro Instrumentation & Telemetry Systems, Inc.
5404 Coal Ave., S. E., Albuquerque, New Mexico 87108

Altair 8800c

Peter Schmidl, 8.3.2025

Agenda

- Geschichte
- Hardware
- Front Panel Programmierung (Showcase)
- Booten von CP/M 2.2 (Showcase)
  - Basic
  - Assembler

Geschichte

- Der Altair 8800 wurde im Dezember 1974 (Jänner 1975) von Micro Instrumentation and Telemetry Systems (MITS) auf den Markt gebracht und wird oft als der Funke bezeichnet, der die Revolution der Personal Computer auslöste. Es war der erste kommerzielle Erfolg für einen Personal Computer (damals Mikrocomputer).

- Dieses bahnbrechende Gerät mit seiner Anordnung von Schaltern und blinkenden LED-Lichtern bot Computerfans ein noch nie da gewesenes Maß an Kontrolle und Anpassungsmöglichkeiten.

- IT Größen wie Bill Gates, Steve Wozniak und Steve Jobs begannen mit diesem Gerät ihre Karriere. Bill Gates (damals noch Student) und Paul Allen entwickelten für dieses Gerät 1975 Altair Basic aus dem später das bekannte Microsoft Basic entstand.

Hardware

- CPU: Intel 8080 bei 2MHz
- Arbeitsspeicher: 256B, erweiterbar auf 64KB
- Bus System: S-100 (extra für den Altair entwickelt)
  - Bis zu 9 Slots für Erweiterungskarten
- Input / Output: Front Panel Switches, LED’s, Kassetten, Floppy, Papierstreifen
  hatte anfangs keine Grafik und musste per Terminal (zB VT100 …) bedient werden
- OS: anfangs keines und dann CP/M von Digital Research

- MITS CPU Card (mit Intel 8080)
- Front Panel Interface Board
- FDC+ (Floppy Controller Card, 64KB RAM und 8K PROM)
- 88-2SIOJP (Dual Serial Card)

Front Panel Programmierung

- Examine: auslesen von Adresse und Anzeigen der Daten
- Deposit: speichern von Daten an die angegebene Adresse
- Stop/Reset: Stoppt jegliche Operationen und bringt das Gerät in einen stabilen Zustand
- zB Speichern von 111 (7 dezimal) auf Adresse 0
  Examine 0 (setzen der Adresse und auslesen) & Deposit mit 111 (Switches)

Step	Address (	8080-Mnemonic	Z80-Mnemonic	Bit Pattern	Octal	Hex	Explanation (Z80 nomenclature)
0.	0000h	LXI H	LD HL,0000h	00 100 001	041	$21	Load HL with 0000h
	0001h	(address)		00 000 000	000	$00	
	0002h	(address)		00 000 000	000	$00	
1.	0003h	MVI D	LD D,80h	00 010 110	026	$16	Load D with 80h
	0004h	(data)		10 000 000	200	$80	
2.	0005h	LXI B	LD BC,000Eh	00 000 001	001	$01	Load BC with 000Eh
	0006h	(data)		00 001 110	016	$0E	
	0007h	(data)		00 000 000	000	$00	
3.	0008h	LDAX D	LD A,(DE)	00 011 010	032	$1A	Load A with the byte at (DE)
4.	0009h	LDAX D	LD A,(DE)	00 011 010	032	$1A	Load A with the byte at (DE)
5.	000Ah	LDAX D	LD A,(DE)	00 011 010	032	$1A	Load A with the byte at (DE)
6.	000Bh	LDAX D	LD A,(DE)	00 011 010	032	$1A	Load A with the byte at (DE)
7.	000Ch	DAD B	ADD HL,BC	00 001 001	011	$09	Add BC to HL
8.	000Dh	JNC	JP NC,0008h	11 010 010	322	$D2	Jump to 0008h if no carry
	000Eh	(address)		00 001 000	010	$08	
	000Fh	(address)		00 000 000	000	$00	
9.	0010h	IN	IN A,(FFh)	11 011 011	333	$DB	Input from port FFh into A
	0011h	(address)		11 111 111	377	$FF	
10.	0012h	XRA D	XOR D	10 101 010	252	$AA	A = A XOR D
11.	0013h	RRC	RRCA	00 001 111	017	$0F	Rotate A right with carry
12.	0014h	MOV D,A	LD D,A	01 010 111	127	$57	Load D with A
13.	0015h	JMP	JP 0008h	11 000 011	303	$C3	Jump to 0008h unconditionally
	0016h	(address)		00 001 000	010	$08	
	0017h	(address)		00 000 000	000	$00	

„Kill the Bit“ Spiel

STEP	MNEMONIC	BIT PATTERN	OCTAL EQUIVALENT
0.	LDA	00 111 010	0 7 2
1.	(address)	10 000 000	2 0 0
2.	(address)	00 000 000	0 0 0
3.	MOV (A->B)	01 000 111	1 0 7
4.	LDA	00 111 010	0 7 2
5.	(address)	10 000 001	2 0 1
6.	(address)	00 000 000	0 0 0
7.	ADD (B+A)	10 000 000	2 0 0
8.	STA	00 110 010	0 6 2
9.	(address)	10 000 010	2 0 2
10.	(address)	00 000 000	0 0 0
11.	JMP	11 000 011	3 0 3
12.	(address)	00 000 000	0 0 0
13.	(address)	00 000 000	0 0 0

Store number one to	10 000 000
Store number two to	10 000 001
RESET -> RUN -> STOP	
EXAMINE (read Result)	10 000 010

Addition von zwei Zahlen

CP/M

DIGITAL RESEARCH

- Ursprünglich Control Program/Monitor und später Control Program for Microcomputers
- Disketten Betriebssystem
- Organisiert Dateien auf einem magnetischen Speichermedium,
- lädt Programme und führt sie aus.
- Quasi Standard Betriebssystem der 1970er / frühe 1980er Jahre
- Entwickelt von Gary Kildall (Digital Research) im Jahr 1974 für 8080/85 Systeme
- Wurde erst mit der Präsentation des ersten IBM PC’s (1981) und seines OS MS-DOS obsolet

Booten nach CP/M

- 88-2SIOJP
  - Einstellen des Jumpstarts zu FF00 (=Disk Boot Loader)
  - Ausschalten des PROM
  - Verbinden eines Terminals zum COM Port der Karte
- PROM der FDC+ Karte aktivieren
- FDC+ Serial Drive Server starten und CP/M 2.2b laden
- RUN am Front Panel drücken => CDBL lädt die Diskette (entweder reales Diskettenlaufwerk oder ein geladenes dsk Image vom FDC+ Server)

FDC+ PROM Content

FF00: Combined Disk Boot Loader (CDBL)
FE00: Altair Multi-Boot Loader (MBL)
FD00: Altair Turnkey Monitor (TURMON)
FC00: Intel Hex file loader
F800: Altair Monitor (ALTMON)

```
56K CP/M 2.2b v2.3
For Altair 8" Floppy

A>dir
A: L80      COM : LADDER   COM : ED       COM : ASM      COM
A: DUMP     COM : XSUB     COM : PCGET    COM : LS       COM
A: SUBMIT   COM : LOAD     COM : SURVEY   COM : VIEW     COM
A: LADDER   DAT : LUNAR    BAS : M80      COM : MAC      COM
A: MBASIC   COM : PIP      COM : STAT     COM : DDT      COM
A: MOVCPM8  COM : NSWP     COM : SYSGEN   COM : TEST     BAS
A: ACOPY    COM : OTHELLO  COM : STARTRK  BAS : TICTAK   BAS
A: WM       COM : WM       HLP : CRC      COM : PCPUT    COM
A: AFORMAT  COM : STARINS  BAS : IOBYTE   TXT
A>
```

Basic

- **MBASIC** (Basic Interpreter)

- Hello World:

```
10 PRINT „Hello World“
20 GOTO 10
```

  save „HELLO.BAS“ => speichert das Listing in die Datei HELLO.BAS
  save „HELLO.BAS“,A => speichert in ASCII (lesbarer Text, nötig für BASCOM)

- **BASCOM** (Basic Compiler von Microsoft)
  => kompiliert Basic Files in ausführbare COM Dateien

  - zB *HELLO,HELLO=HELLO => erzeugt ein Object (.REL) und Listing (.PRN) file

  - L80 HELLO,HELLO/N/E => ergibt ein HELLO.COM (Runtime BRUN.COM notwendig)

Assembler

- **ASM** (CP/M Assembler)
  - ASM HELLO erzeugt ein Listing (PRN) und Intel Hex File (HEX)
  - LOAD HELLO erzeugt aus dem HEX File ein ausführbares HELLO.COM
- **DDT** (Dynamic Debugging Tool)
  DDT TEST.COM => disassembled HELLO.COM
  D ... zeigt den Speicher als HEX und ASCII an
  L ... zeige den Speicher als Assembler Mnemonics